

## **Team038-Inside Out Project Proposal:**

**Title:** CourseCompass

### **Project Summary:**

CourseCompass attempts to combine three beloved applications used by students on UIUC campus: GPA Disparity, Rate My Professor, and Course Explorer. Instead of having to navigate through all three sites to cross-examine the GPA distribution for a certain professor in a certain course across sections, CourseCompass allows students to receive all that information from one place. CourseCompass will provide a user-friendly interface through which students can search up specific courses, subjects, professors and course sections to examine the distribution of grades, ratings of professors from other students, and information about courses and offered sections to make well-informed decisions about their schedule. Using this information, they will also be able to make a tentative course schedule, on our application itself.

### **Description:**

The main problem that we want to solve is the inconvenience students face of having to cross check information from three different sources while building schedules for the semester. This application hopes to reduce the uncertainty and lack of information students face when choosing their schedules for upcoming semesters. By offering detailed reviews, course information and grade distributions, all in one place, students can gain insight into different teacher's teaching styles, class structure, and course rigor to make educated decisions about how to structure their schedule.

As stated before, our project is a combination of three applications that already exist and are used by students currently. Although these applications already exist, by combining them into one application, we hope to alleviate the trouble students face having to check all three web pages for information about their courses, offered sections, and prior student's experiences with certain teachers by offering them all in one web page. By centralizing all of this information, we aim to lessen the stress a student faces during the course planning/scheduling process.

### **Creative Component:**

One creative component that might improve our project is a course predictor. For example, a student could input a few parameters (difficulty level, subject, preferred average gpa, etc.) and our website could return the top 3-5 recommendations that fit that criteria. If we wanted to include this feature, we would have to build a sorting algorithm that parses through the courses and their attributes to find which courses fulfill requirements entered by the user.

To add on to the creative component, we could also categorize the recommendations based on general education requirements. For example, if a student was looking for a general education

course for humanities, the predictor would recommend courses specific to that general education requirement. This is an extra component because there are many different subjects and classes that fall under the larger humanities category, so allowing the student to quickly identify the classes they want makes it much more convenient for them.

### **Usefulness:**

Our chosen application is useful because it aims to make course registration more convenient for students. As students select courses for an upcoming semester, they tend to look through the grade distributions for their courses as well as the professor's ratings in order to curate their ideal schedule. However, this information is currently scattered among various resources. We aim to reduce this inconvenience by having all the information available in one place.

Three sites already exist that contain information similar to our application: Wade's GPA disparity visualizations, RateMyProfessor, and Course Explorer. Individually, the components of these three sites make up the bulk of our application; the difference is that we are hoping to combine their information to make it more convenient for students to navigate. This application is especially useful for students that have to choose courses that are not required for their major (i.e. general education courses).

### **Realness:**

Three data sources that we are planning on using are the UIUC Course Catalog dataset, UIUC GenEd dataset and the UIUC GPA dataset. The UIUC Course Catalog dataset includes all courses that were offered in the Spring 2024 semester (data source updates as they come out for each semester). The dataset has 27 columns including Year, Term, Subject, Number, Name, etc and has 11904 observations (rows). The GPA dataset includes all courses that were offered up until Summer 2023 and their grade distributions across all primary instructors. The dataset has 22 columns, including Year, Term, Subject, A+ - F, Primary Instructor, etc and has 69070 observations (rows). The GenEd dataset includes all GenEd courses that were offered in the Spring 2023 semester. The dataset has 11 columns including Year, Term, Course Title, GenEd categories and has 1061 observations (rows). These datasets are collected from UIUC's databases and made public. They are both provided in csv format.

Each of these datasets will be used in our project to satisfy a certain feature(s). The UIUC Course Catalog dataset will be used as a basis for the courses offered in a certain semester. The UIUC GPA dataset contains the grade distributions for courses offered at UIUC dating all the way back to 2010. The UIUC GenEd dataset contains all the GenEd requirements that certain courses fulfill. Combined, these datasets can be used to easily search up whether a certain course offered in a given semester fulfills any GenEd requirements and what the past grade distributions are for the teachers leading the course. Using this information, students can easily curate a schedule to their own taste.

Dataset Links:

<https://github.com/wadefagen/datasets/tree/master/gpa>

<https://github.com/wadefagen/datasets/tree/master/course-catalog>

<https://github.com/wadefagen/datasets/tree/master/geneds>

### **Functionalities:**

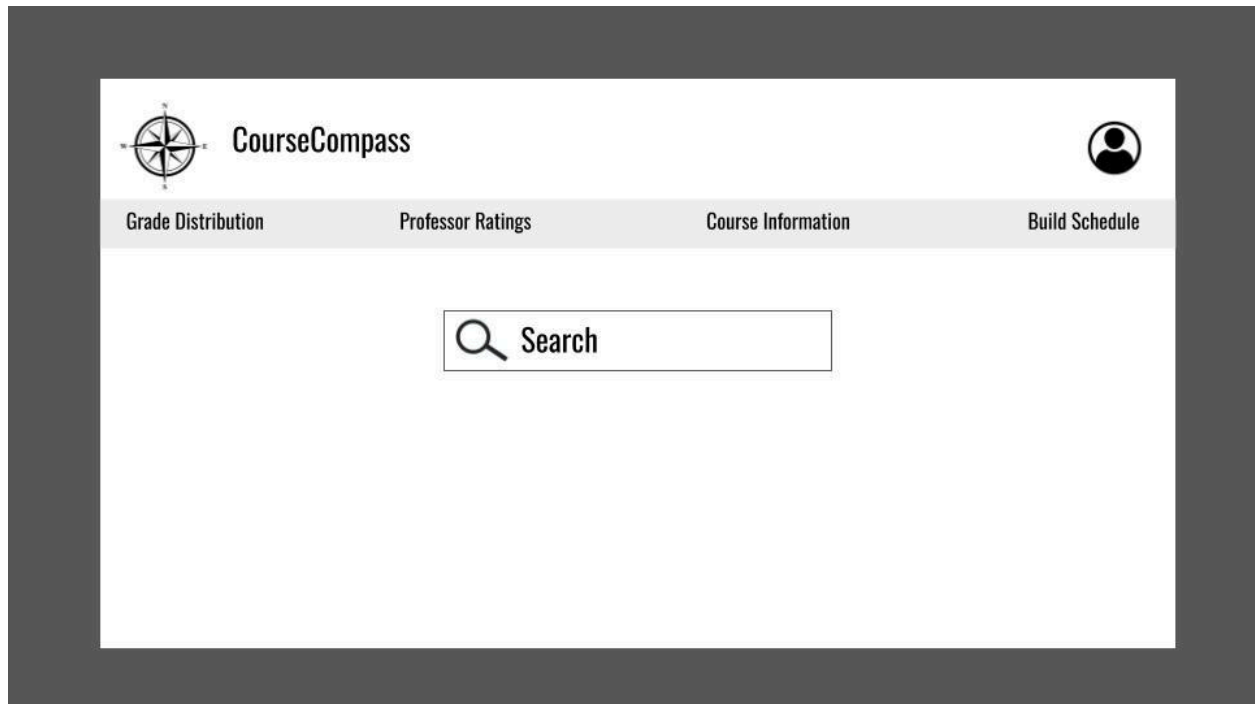
Since our application aims to allow students/users to be able to search for, write their own ratings, and build a potential schedule, there are many opportunities for them to interact with our web page.

### **CRUD Operations:**

1. **Create Operations:** In our application, the "create" operation facilitates the addition of new records to the relevant tables. This process is executed through SQL INSERT queries. For instance, users engage in the creation of their profiles, resulting in the insertion of their information into the Users table. Similarly, users can contribute new ratings for professors they've encountered, leading to the insertion of these ratings into the appropriate table. Additionally, users can edit their tentative schedules by adding new courses, which are then inserted into the respective table for schedules.
2. **Delete Operations:** We will involve the use of SQL DELETE queries to eliminate records from the tables. Users can delete their profiles, which triggers the deletion of their record from the Users table along with any associated ratings or schedule entries. Additionally, users can delete specific ratings as needed.
3. **Update operation:** We can use the UPDATE operation to update a professor's average rating each time a student submits a new rating for that professor. We can also use this operation when users want to update portions of their user profile or alter ratings that they've already posted.
4. **Lastly,** users are able to "search for" courses, subjects, or professors that they are interested in to find which class sections they would be interested in and add it to their profile.

### **UI Mockup:**

**For the proposed UI mockup for CourseCompass, we plan on having a landing page exhibiting a search bar that allows users to input specific courses, subjects, or professors they wish to explore. The navigation bar will provide access to different sections, including "Grade Distribution," "Professor Ratings," "Course Information," and "Build Schedule."**



### **Work Distribution:**

Everyone will do different parts of the backend. Shriya will work on the first steps involving integrating the datasets into a SQL database. Ashwini will work on joining all the tables from the three datasets, using course numbers and professors as common values. Niv will work on the functionality that allows the users to make their own profiles and add their ratings to the database. Anisha will work on creating the searching functionality with SQL queries. Niv and Ashwini will also work on the schedule builder tool. We will all work on the needed transactions, stored procedures, triggers, and constraints needed to satisfy the features/requirements of our application that we listed above. We will also all work on the required testing needed to make sure our features are working as expected.

Anisha and Shriya will work on the front-end, which will most likely be done with React and JavaScript. Ashwini and Niv will work on tying the front-end and back-end.